Language Identification

Joaquín Mestanza
Lucero Guadalupe Fernandez
Instituto Tecnológico de Buenos Aires

***Abstract*—En el presente proyecto se realizó la identificación de tres idiomas: español, inglés, y alemán mediante una red neuronal convolucional (CNN). Es el objetivo del proyecto obtener los features que mejor solucionan el problema y maximizar la eficacia de clasificación de la red; para cumplir este objetivo, se preprocesaron los datos para obtener los features a partir de las muestras del dataset como la posterior entrada a la red y se analizaron diferentes arquitecturas de la misma.**

I. Introducción

El presente trabajo se centra en la utilización de una red neuronal convolucional (CNN). Estas redes tienen su fortaleza en reconocer patrones visuales, y son especialmente útiles para clasificar imágenes. Lo que se propone entonces es obtener los espectrogramas de los audios en diferentes idiomas y alimentar con eso a la red pensando al espectrograma como una imagen. La metodología propuesta puede verse en la Figura 1, de manera que la red pueda clasificar entre ellos.

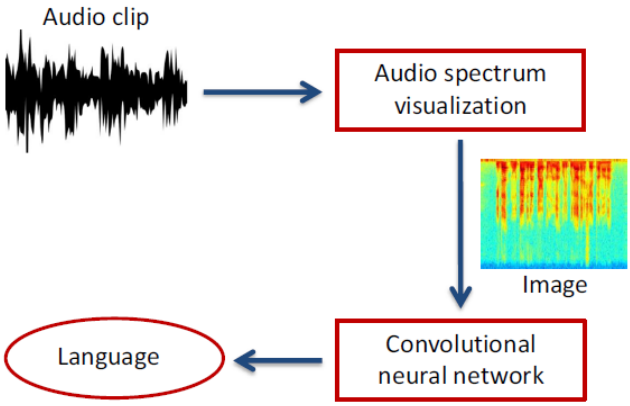

Fig. 1. Metodología propuesta

II. Dataset

El dataset es una parte muy importante en todo proyecto, ya que es el punto de partida. Frente a esto hay dos posibles enfoques: utilizar un gran dataset, donde haya audios de speakers de diferentes edades, sexo y acento. La segunda posibilidad es utilizar un pequeño dataset y aumentar la cantidad de speakers 'únicos' con *data augmentation*. El segundo enfoque fue el que se tomó y la razón por la que se escogió el dataset que se detallará a continuación obtenido a partir de LibriVox.[1] [2]

[1] https://librivox.org/
[2] https://github.com/tomasz-oponowicz/spoken_language_dataset/

Las características principales del dataset se enumeran a continuación:

- Contiene grabaciones en tres idiomas: español, inglés y alemán y cada audio dura 10 segundos.
- El dataset se encuentra balanceado, esto es, una cantidad igual de samples para todos los idiomas y una igual cantidad de speakers masculinos y femeninos.
- Los speakers presentes en test no se encuentran en train, esto es para evitar que el modelo 'memorice' al speaker.
- El dataset originalmente contenía 90 speakers únicos, pero mediante data augmentation se varió el pitch (8 niveles diferentes) y la velocidad de las samples (8 niveles de velocidad), con lo que se obtuvieron finalmente 1530 speakers únicos.
- El dataset de train contenía 73080 samples que debieron ser reducidas por falta de recursos, léase memoria RAM, lo que da paso a la división final del dataset, de lo que se hablará a continuación.
- Originalmente el dataset estaba dividido en 73080 samples en train y 540 en test. Como se comentó en el ítem anterior, hubo que modificarlo por limitaciones técnicas, y además se dividió en train, validation y test. La organización final fue de 3288 samples en train, 366 en validation y 540 samples en test.

III. Preprocesamiento de Datos

Al momento de procesar los datos para extraer los features de las muestras se tienen dos posibles enfoques. En primer lugar, los MFCCs (Mel-Frequency Cepstral Coefficients) fueron muy populares como features por mucho tiempo, pero recientemente los bancos de filtros se volvieron mucho más populares por diversas razones que introduciremos en breve.

El cálculo de los bancos de filtros y de los MFCCs involucran casi el mismo procedimiento, donde al obtener los bancos de filtros se pueden obtener los MFCCs con unos pocos pasos más.

A. Pre-énfasis

El primer paso a seguir es aplicar un filtro pasa altos de manera de amplificar las frecuencias altas. Esto es útil por varias razones: (1) balancea el espectro dado que las altas frecuencias suelen ser de menor magnitud comparada con frecuencias más bajas, (2) evita errores numéricos al calcular la FFT, y (3), mejora la relación señal a ruido (SNR).

El filtro pre-énfasis se aplica a la señal $x(t)$ usando un filtro de primer orden como se muestra en la siguiente ecuación:

$$y(t) = x(t) + \alpha x(t-1)$$

donde un valor típico para α es 0.95 o 0.97.

Los efectos de este filtro tienen un efecto modesto en los sistemas actuales, ya que efectos similares se pueden obtener normalizando, excepto al momento de evitar errores numéricos.

B. Framing

Luego de aplicar el filtro de pre-énfasis, lo que se realiza es dividir la señal en pequeños frames de corta duración. Esto es porque las frecuencias de la señal cambian con el tiempo y carece de sentido realizar la FFT a toda la señal completa. Por esto asumimos que la señal es estacionaria durante un período de tiempo, durante un frame, para poder obtener una aproximación de las frecuencias de la señal en el tiempo concatenando frames adyacentes.

Se eligieron frames de duración entre 20ms y 40ms con aproximadamente 50% de overlap entre frames consecutivos, en particular, frames de 25ms de duración y 15ms de overlap.

C. Ventaneo

A continuación de dividir la señal en frames, se le aplica un ventaneo con ventana de Hamming a cada frame. Una ventana de Hamming tiene la forma de la Figura 2 y cumple que:

$$w[n] = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right)$$

donde $0 \leq n \leq N-1$, con N el tamaño de la ventana.

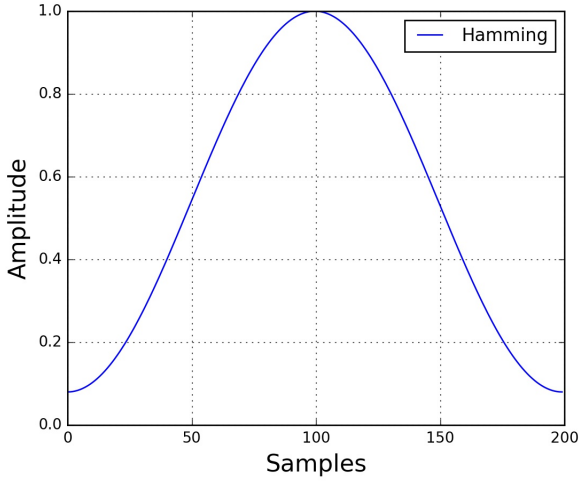

Fig. 2. Ventana de Hamming

La principal razón para ventanear es para reducir el leakage espectral.

D. FFT y Power Spectrum

Como paso siguiente,se realiza la transformada de Fourier de N puntos, aplicando la STFT, para N=512, para calcular el espectro a cada frame. Luego, se realiza la estimación del espectro de potencia calculando el periodograma según:

$$P = \frac{|FFT(x_i)|^2}{N}$$

donde x_i es el frame i-ésimo de la señal x.

E. Filter Banks

El último paso para calcular los bancos de filtros es aplicar filtros triangulares, un valor típico y el que se usó son 40 filtros en la escala Mel a la estimación del espectro de potencia para extraer las bandas de frecuencia.

Cada filtro del banco de filtros es triangular, como ya se mencionó y tiene respuesta unitaria en la frecuencia central y decrece hasta la mitad del filtro adyacente, como se puede observar en la figura siguiente:

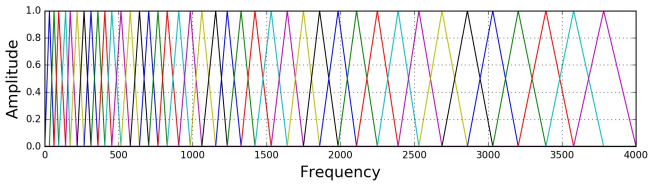

Fig. 3. Mel Filter Banks

Luego de aplicar el banco de filtros al espectro de potencia, es decir, al periodograma de la señal se obtiene el siguiente espectrograma:

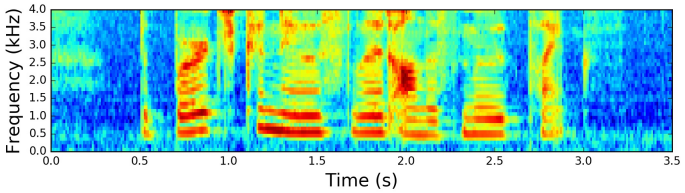

Fig. 4. Mel Filter Banks

Como aclaración, si los features a utilizar son el banco de filtros lo único que resta es normalizar. El paso siguiente explica cómo obtener los MFCCs y luego se discuten los features a utilizar.

F. MFCC

Los coeficientes del banco de filtros están altamente correlacionados, lo que podría complicar la situación de algunos algoritmos de machine learning. Entonces se debe aplicar la DCT (Discrete Cosine Transform) para descorrelacionar los mismos. Cabe aclarar que algunos son descartados y se mantienen los coeficientes 2-13. A continuación y como paso final solo resta normalizar, lo que da paso a la siguiente figura:

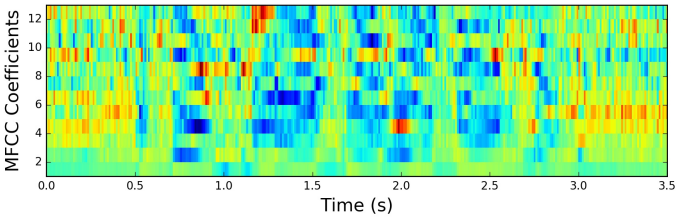

Fig. 5. Mel-Frequency Cepstral Coefficients

Como ya se mencionó, los MFCCs se utilizaron por mucho tiempo, cuando prevalecía el uso de los Gaussian Mixture Models y los Hidden Markov Models (GMMs-HMMs) y junto a los MFCC era la manera de realizar reconocimiento de voz. Cuando surgieron las técnicas de Deep Learning, las redes neuronales eran menos susceptibles a entradas altamente correlacionadas y carecía de sentido realizar la DCT para descorrelacionar los coeficientes del banco de filtros. Con todo esto, dado que se utiliza una red neuronal convolucional se decidió por utilizar el espectrograma obtenido del banco de filtros aplicado al periodograma como entrada de la red.

IV. Modelo de red

Para poder encontrar el modelo que finalmente se adoptó, se intentaron varias configuraciones. Se investigó en primer lugar la famosa arquitectura AlexNet dado que fue una revolución en su tiempo para los problemas de redes convolucionales con identificación de imágenes. La misma tiene la particularidad de que a medida que se va ganando profundidad en la red respecto del input, se va ganando más detalle respecto a la imagen, dado que se agregan filtros y se disminuye el tamaño del kernel de las redes convolucionales. Además se utiliza un maxpooling dado que éste tiene la característica de quedarse con lo que más resaltó en el pool, lo cual podemos asociar si se quiere (cuando el número máximo es muy alto respecto de los demás) a un spike neuronal.

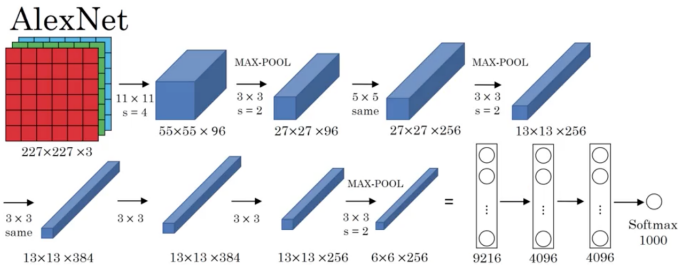

Fig. 6. Arquitectura AlexNet

Por otro lado, se investigó la arquitectura del trabajo ”Deep learning for spoken language identification: Can we visualize speech signal patterns?” [1], la cual estaba más relacionada con el objetivo del presente trabajo.

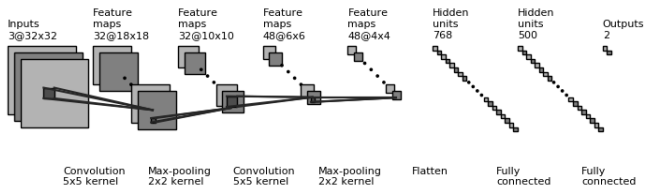

Fig. 7. Arquitectura del trabajo ”Deep learning for spoken language identification: Can we visualize speech signal patterns?”

Tanto en AlexNet como en la última arquitectura presentada, se realiza max pooling con kernels de menor tamaño (respecto al de la convolución).

Ambas tienen capas densas antes de aplicar la función softmax.

Con estos dos modelos en mente, se procedió a hacer una primera arquitectura. En esta nueva arquitectura, se encontró que si bien los resultados eran aceptables logrando aproximadamente accuracy de 100% en train y 84% de accuracy en validation, no son de los mejores resultados encontrados en la práctica utilizando las mismas técnicas. Es por eso que se regularizó de distintas formas, conservando el accuracy en train y llegando a valores de accuracy más altos (alrededor de 87% en validation) pero no se mostraba un cambio tan significativo y se podía ver una brecha grande entre train y validation. Luego de los intentos de mejorar los resultados mediante regularización tradicionales, se llegó a la conclusión de que se tenía que tratar de un cambio más radical, un cambio en la arquitectura.

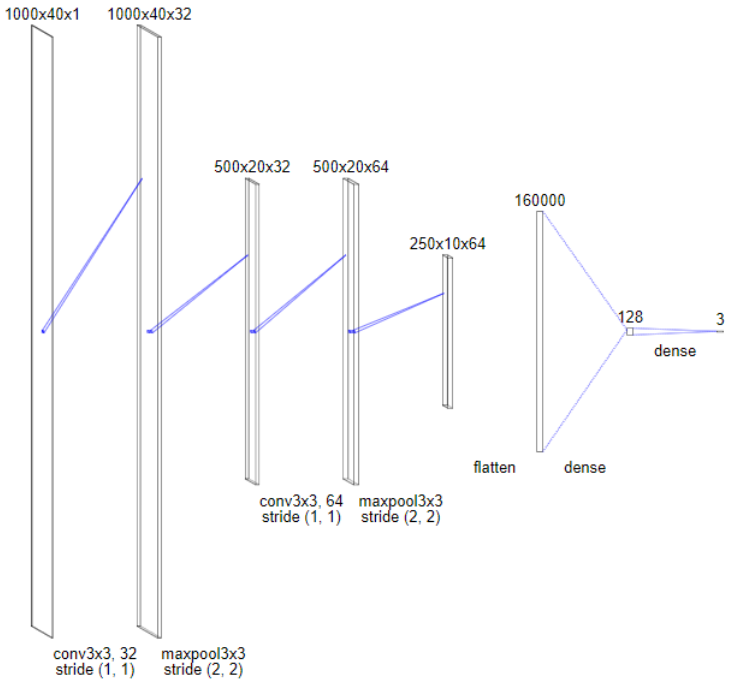

Fig. 8. Primera arquitectura propuesta

Lo que en primer lugar se pensó es que la red no generalizaba lo suficientemente bien por lo que se intentó tratar de bajar los parámetros disminuyendo el tamaño de la red. Lo que se pudo observar es que no mejoraban los resultados, sino que empeoraban. Es por esto que se tuvo un indicio de que el enfoque tenía que ser más orientado a Deep Learning, por lo cual se optó por añadir una serie de capas las cuales puedan dar más detalles de la imagen a la red, teniendo el cuidado de no aumentar los parámetros de la red desmesuradamente.

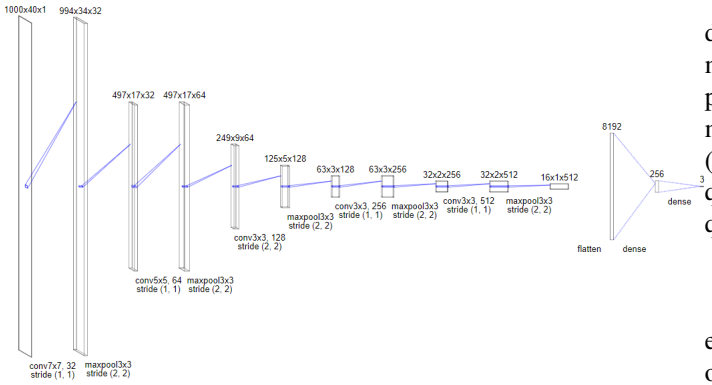

Fig. 9. Arquitectura final

Una vez obtenida la nueva arquitectura, se realizó el entrenamiento correspondiente y finalmente se pudo lograr el objetivo que se había planteado, llegando a un 100% de accuracy en train y 99.45% de accuracy en validation.

V. Resultados

A continuación se muestra la matriz de confusión, que permite visualizar la performance de la red. La diagonal representa las instancias (en porcentaje) predichas correctamente para cada clase, mientras que las demás celdas se trata de los casos donde no acertó.

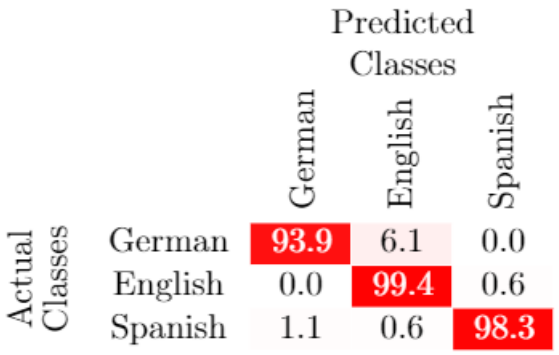

Fig. 10. Matriz de Confusión

El hecho de haber podido obtener resultados de este calibre, están dados por la arquitectura, la elección del optimizador (Adam) y las técnicas de regularización utilizadas (Dropout y LearningRateScheduler).

En este caso, puede resultar interesante observar cuándo la red se equivoca, para ganar un poco de intuición por sobre lo que está pasando detrás del aprendizaje. Los casos que tomaremos como ejemplo para este análisis son los de Inglés y Alemán.

En el caso que predice que es inglés dado que es alemán integra el 6.1% de los casos donde era alemán. En el caso que predice que es alemán dado que es inglés representa (por redondeo) el 0% de los casos donde era inglés.

Para poder explicar este hecho, se tiene en cuenta la cantidad de fonemas vocálicos en principio, dado que estos son los que más energía tienen y más presente están en un idioma. Es posible argumentar que la distribución de fonemas puede ser muy distinta y en ese caso, perdería validez este argumento (es decir, no sólo depende de la cantidad). Es por esta razón que se eligió alemán e inglés a modo de comparación, dado que ambos son de origen germánico.

Fonemas vocálicos: aleman 16, inglés 12, español 5.

El razonamiento parte de la premisa de que si hay una equivocación en la predición, es más probable que se equivoque con el idioma que posee inmediatamente menos fonemas vocálicos.

Este razonamiento se corresponde con que el alemán se confunde con el inglés con más probabilidad que en el caso en el que confude inglés con el alemán. De hecho, también valida el hecho de que el inglés se confunda más con español.

Con esto en mente, se puede realizar una observación similar con el Español, pero en este caso utilizamos los fonemas consonánticos ya que el español es el que menos fonemas vocálicos posee y no podemos utilizar el argumento de los fonemas vocálicos.

Fonemas consonánticos: inglés 22, español 19, alemán 16-20.

De forma análoga, se observa que español se confunde más con el idioma que tiene menos fonemas consonánticos, que en este caso se asume que se trata del alemán.

También se calculó la F1 Score, dada por:

$$F1Score = 2 * \frac{precision * recall}{precision + recall} = 0.97$$

siendo la *precisión* la cantidad de resultados positivos correctos dividida por la cantidad de todos los resultados positivos devueltos por esa categoría. Mientras que el *recall* es la cantidad de resultados positivos correctos dividida la cantidad total de las muestras relevantes (todas las que deberían haber sido identificadas como positivas).

Además se graficó la accuraccy tanto para el set de train como para el de validación y también la función de costo (loss), que se muestran en las Fig. 11 y 12. Se puede observar que convergen alrededor de los 20 epochs.

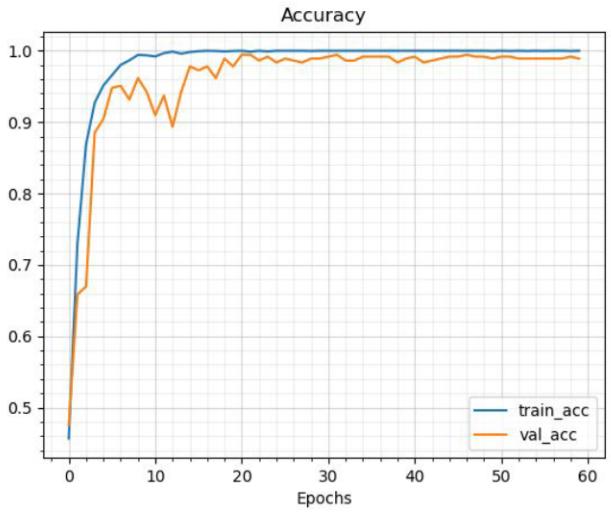

Fig. 11. Accuracy de train vs validation

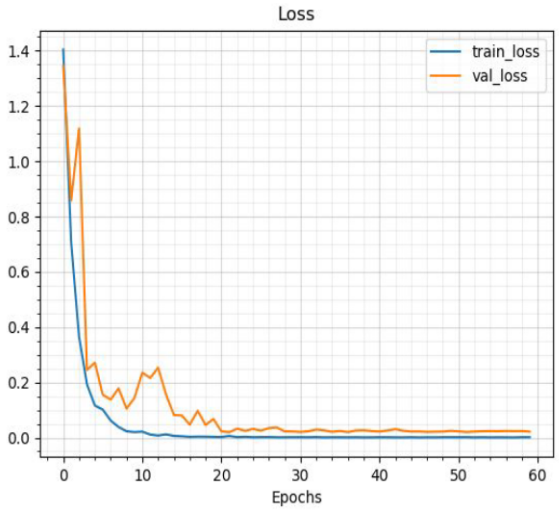

Fig. 12. Loss, función de costo de train vs validation

VI. Conclusiones

Volviendo al análisis de resultados, si bien las hipótesis parecen tener correspondencia con lo que sucede, puede aparecer un caso donde la hipótesis de cantidad de fonemas no resulta general. Este es un problema que se suele tener en el ámbito de redes neuronales y machine learning cotidianamente. Este tipo de análisis se encuentra en una relación estrecha con la conocida frase "correlación no implica causalidad". Es decir, siempre se debe tener en cuenta del contexto en el que se está dando.

Como se mencionó en la sección anterior, se obtuvieron resultados muy favorables, aproximados al estado del arte actual. Se logró maximizar la eficacia de la red obteniendo como features el banco de filtros y descartando los MFCCs y se pudo aprovechar el uso de la red neuronal convolucional al utilizar como entrada el espectrograma obtenido a partir del banco de filtros.

Futuros trabajos podrían incluir grabaciones y audios con ruido de fondo y reverberación para asemejar más un ambiente cotidiano y normal.

References

[1] H. Mukherjee, S. Ghosh, S. Sen, O. Sk Md, K. C. Santosh, S. Phadikar, K. Roy., "Deep learning for spoken language identification: Can we visualize speech signal patterns?", 2019.

[2] C. Bartz, T. Herold, H. Yang and C. Meinel, "Language Identification Using Deep Convolutional Recurrent Neural Networks", 2017.

[3] H. M. Fayek, "Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between", 2016 [Accesed June 2020], https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html